

# Technical Disclosure Commons

---

## Defensive Publications Series

---

October 2020

## Enhancing Virtual Assistant Clarification Questions By Predicting Missing Parameters

Matthew Sharifi

Victor Carbune

Follow this and additional works at: [https://www.tdcommons.org/dpubs\\_series](https://www.tdcommons.org/dpubs_series)

---

### Recommended Citation

Sharifi, Matthew and Carbune, Victor, "Enhancing Virtual Assistant Clarification Questions By Predicting Missing Parameters", Technical Disclosure Commons, (October 29, 2020)  
[https://www.tdcommons.org/dpubs\\_series/3731](https://www.tdcommons.org/dpubs_series/3731)



This work is licensed under a [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/).

This Article is brought to you for free and open access by Technical Disclosure Commons. It has been accepted for inclusion in Defensive Publications Series by an authorized administrator of Technical Disclosure Commons.

## **Enhancing Virtual Assistant Clarification Questions By Predicting Missing Parameters**

### **ABSTRACT**

User interactions with virtual assistants often involve commands that require the user to specify one or more relevant parameters. If one or more of the needed parameters is missing, the interaction can turn into an extended dialog involving multiple turns for the virtual assistant to obtain the necessary parameters from the user. This disclosure describes techniques to enhance user interactions with a virtual assistant in cases that require clarification questions to obtain one or more required parameters missing from the original request. Clarification questions are refined based on user-permitted contextual information and can include suggestions of the most likely parameter value(s) as part of the question.

### **KEYWORDS**

- Virtual assistants
- Clarification questions
- Multi-turn dialog
- Missing command parameters
- Machine learning models
- Parameter prediction
- Context awareness

### **BACKGROUND**

Voice-based virtual assistants enable people to engage in natural conversational interaction to obtain information and/or perform actions. Users interact with virtual assistants via various devices, such as smartphones, smart speakers, smartwatches, etc. These interactions often involve requests related to productivity, such as commands for sending messages, setting

timers, etc. Such commands typically require the user to specify one or more relevant parameters. For instance, a command to send a message might require that the user request specify the recipient, the subject, and the content of the message. If one or more of the needed parameters is missing, the user's interaction with the virtual assistant can turn into an extended dialog involving multiple turns for the virtual assistant to obtain the necessary parameters from the user. For example, if the user asks a virtual assistant to "send a message to Alice," the content to be relayed would need to be obtained by asking the user a clarification question, such as "what would you like to say to Alice?" to receive a response such as "I'm running late; will join the meeting in 5 minutes." There may also possibly be a follow-up turn to confirm that the virtual assistant correctly received the spoken message, if the virtual assistant speech-to-text conversion of the user's speech is not of sufficiently high confidence.

## DESCRIPTION

This disclosure describes techniques to enhance user interactions with a virtual assistant in cases that require clarification questions to obtain one or more required parameters missing from the original request. The described techniques refine the clarification questions by suggesting the most likely parameter(s) as part of the question. The most likely parameter(s) can be determined based on relevant information from the user's context, obtained with permission.

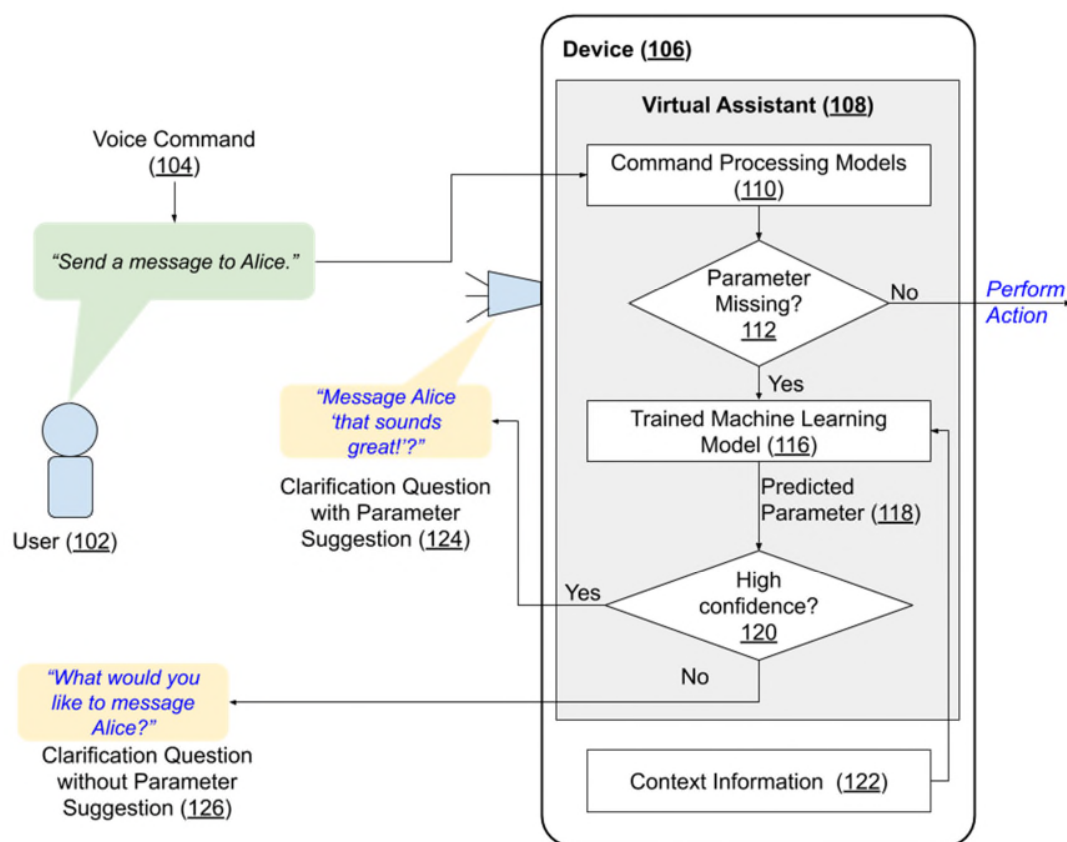
For example, if a user asks a virtual assistant to "send a message to Alice," while running late for a meeting with Alice, the clarification question can be "message Alice that you are running late?" Such a clarification question includes a suggestion for the message content parameter, thus enabling the user to complete the action by responding with a "Yes" rather than needing to specify the entire message content in response to a generic clarification question such as "what would you like to tell Alice?"

Users can issue a spoken command to the virtual assistant as they normally do, with the virtual assistant being invoked using interaction techniques such as an activation hotword, gesture, button press, etc. The command is processed using Automated Speech Recognition (ASR) and is interpreted using Natural Language Understanding (NLU) techniques. If the results of the interpretation indicate that the command includes sufficient parameters, it is carried out as usual without asking the user any clarifications question. For example, such a command may include “Play my workout playlist” where “workout playlist” is a parameter that refers to a particular playlist in the user’s music collection.

On the other hand, if command interpretation indicates that one or more parameters required to carry out the user’s request are missing, a suitable trained machine learning model is employed to predict the likely value(s) for the missing parameter(s) based on accessing user contextual data with user permission. The model predicts a set of parameter values for each missing parameter based on relevant user-permitted contextual information such as message history, calendar entries, routines, time, parameters included in previous commands, etc. For instance, the context relevant for a messaging command could be the previous several messages with the intended recipient and/or calendar entries that involve the intended recipient. Similarly, the context relevant for a timer setting command might be timer values previously requested at similar times and/or during similar activities.

Each predicted parameter value in the set generated by the trained machine learning model is associated with a score indicating the likelihood of it being the parameter intended by the user. If the highest score among the predicted parameter set is above a threshold value, the corresponding predicted parameter value is included as a suggestion in the clarification question pertaining to the parameter missing from the user’s original command. For example, if the

context is that the user is responding to Alice regarding dinner plans and the model prediction indicates that the user's desired reply is highly likely to be 'That sounds great!,' then the clarification question can be "message Alice 'that sounds great!'" instead of the generic "what would you like to tell Alice?" If the user confirms acceptance of the suggestion by answering the clarification question in the affirmative, the command is carried out using the suggested parameter included within the clarification question.



**Fig. 1: Asking for clarification by including a high confidence parameter suggestion**

Fig. 1 shows an operational implementation of the techniques described in this disclosure. A user (102) issues a voice command (104) requesting a virtual assistant (108) on a device (106) to send a message to Alice. The command is processed via the usual command processing

models (110) to determine if it is missing any needed parameter (112). If the required parameters are included within the command the user's requested action is performed.

In the example shown in Fig. 1, the user's command is missing the content of the message the user wishes to send. A trained machine learning model (116) is used to predict the missing parameter (118) using relevant context information (122). If the predicted value is highly likely (120) to indicate the user's intent the clarification question asked to the user to seek the missing parameter includes the predicted parameter suggestion (124). If a high confidence prediction is not available, the user is asked a generic clarification question without any parameter suggestions (126).

If the user finds that the predicted parameter suggestion included in the clarification question is unsuitable, the user has the option to override the suggestion and provide the desired parameter value in response. For instance, the user may respond to a clarification question with "No. Tell Alice that I will not be able to join her for dinner tonight."

With user permission, the described techniques can be applied for refining clarification questions pertaining to predicting missing parameters in any supported virtual assistant commands such as sending messages, setting timers, playing music, making reservations, etc. If the user permits, apart from handling commands with missing parameter(s), the techniques can also be applied in situations where the user's command is received only partially because of external influences, such as noise.

The techniques described above support prediction and inclusion of multiple missing parameters when refining clarification questions with user permission. For instance, a clarification question pertaining to the user's command to play music (e.g., "I'm in the mood for jazz") can include suggestions for the artist and the song, provided both of these missing

parameters can be predicted with high confidence. With user permission, the length of a predicted parameter value can be considered as an additional factor when determining whether the value is suitable for inclusion with a clarification question. For instance, overly long parameter suggestions can be discarded to minimize user intrusion.

The techniques described herein can be implemented within any device that includes virtual assistant capabilities. The trained machine learning models used for ASR, NLU, and missing parameter prediction can be any suitable machine learning model, such as a deep neural network. The ASR and NLU operations required for processing and interpreting the user's commands can be performed locally on the device and/or externally on the server, if the user permits server-side processing. The various threshold values used in the above operation can be set by the developers and/or specified by the user and/or determined dynamically at runtime.

Implementation of the described techniques can minimize the need for users to provide command parameters explicitly, thus saving time. Moreover, the techniques reduce the likelihood of the user's parameter specification being misunderstood since the user needs only to confirm or decline the suggested parameter included in the enhanced clarification questions. These benefits serve to enhance the user experience (UX) of voice-based virtual assistants.

Further to the descriptions above, a user may be provided with controls allowing the user to make an election as to both if and when systems, programs or features described herein may enable collection of user information (e.g., information about a user's calendar, prior message history, media preferences, interactions with a virtual assistant, social network, social actions or activities, profession, other preferences, or current location), and if the user is sent content or communications from a server. In addition, certain data may be treated in one or more ways before it is stored or used, so that personally identifiable information is removed. For example, a

user's identity may be treated so that no personally identifiable information can be determined for the user, or a user's geographic location may be generalized where location information is obtained (such as to a city, ZIP code, or state level), so that a particular location of a user cannot be determined. Thus, the user may have control over what information is collected about the user, how that information is used, and what information is provided to the user.

## CONCLUSION

This disclosure describes techniques to enhance user interactions with a virtual assistant in cases that require clarification questions to obtain one or more required parameters missing from the original request. Clarification questions are refined based on user-permitted contextual information and can include suggestions of the most likely parameter value(s) as part of the question. A suitable trained machine learning model is employed to predict the likely value(s) for the missing parameter(s) based on user-permitted contextual information, and the most likely value(s) are included in the clarification question. Implementation of the described techniques can save time by minimizing the need for the user to provide command parameters explicitly and reduces the likelihood of misunderstanding since the user needs to only confirm or decline the suggested parameter.